(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 April 2002 (11.04.2002)

PCT

(10) International Publication Number
WO 02/29717 A2

(51) International Patent Classification[7]: G06T

(21) International Application Number: PCT/US01/30858

(22) International Filing Date: 2 October 2001 (02.10.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/237,311 2 October 2000 (02.10.2000) US

(71) Applicant *(for all designated States except US)*: **THE RESEARCH FOUNDATION OF STATE UNIVERSITY OF NEW YORK** [US/US]; Office of Technology Licensing, S.U.N.Y. at Stony Brook, Stony Brook, NY 11794 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KAUFMAN, Arie, E.** [US/US]; 94 Cedar Drive West, Plainview, NY 11803 (US). **LIANG, Zhengrong** [US/US]; 28 Houghton Boulevard, Stony Brook, NY 11794 (US). **WAX, Mark, R.** [US/US]; 6 East Sanders Street, Greenlawn, NY 11740 (US). **BITTER, Ingmar** [DE/US]; 1456 Stony Brook Road, Stony Brook, NY 11790 (US). **DACHILLE, Frank, IX** [US/US]; 94 Central Avenue, Amityville, NY 11701 (US). **KREEGER, Kevin** [US/US]; 115 Randall Avenue, Port Jefferson, NY 11777 (US). **WAN, Ming** [CN/US]; 649 Chapin Complex, Stony Brook, NY 11790 (US).

(74) Agents: **TANG, Henry** et al.; Baker Botts LLP, 30 Rockefeller Plaza, New York, NY 10112-0228 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/29717 A2

(54) Title: CENTERLINE AND TREE BRANCH SKELETON DETERMINATION FOR VIRTUAL OBJECTS

(57) Abstract: In accordance with the present invention, a method for determining a centerline through a region of interest in a 3D image dataset is provided. The method includes identifying the boundaries of the region of interest and identifying the endpoints of the region of interest. For those points within the boundaries, a penalty value which is a function of the proximity of the point to a boundary is determined. A centerline is then identified by the path connecting the endpoints which has the minimum penalized distance wherein the penalized distance reflects the actual accumulated pathlength and the penalties associated with the points along the path. From the centerline, branches of a complete skeleton can be established by determining branch endpoints and then finding the minimum penalized distance from each endpoint the centerline or another intersecting branch.

BNSDOCID: <WO_____0229717A2_I_>

CENTERLINE AND TREE BRANCH SKELETON DETERMINATION FOR VIRTUAL OBJECTS

SPECIFICATION

The subject matter of this application was funded in part by the National Institute of Health, contract number CA79180 and the U.S. Office of Naval Research, grant number N000149710402. From these grants, the U.S. government may have certain rights to the invention.

FIELD OF THE INVENTION

The present invention relates generally to computer-based three dimensional visualization and more particularly relates to improved methods for virtual navigation including skeleton generation.

BACKGROUND OF THE INVENTION

In many computer-based visualization systems, it is desirable to be able to define a skeleton or centerline through the object being viewed. Such a skeleton provides a reference path for virtual navigation, such as in virtual colonoscopy. Similarly, accurate length measurements and navigation through other organs, such as the aorta, also require an object skeleton. Of course, the need to define an accurate centerline or tree is not limited to medical imaging. Other fields, such as virtual engineering and architecture would also benefit from accurate methods of defining a skeleton in tree like objects or generating a centerline in linear objects in various visualization applications.

Various techniques are known for generating a center line for an object being visualized. For example, in the virtual examination of organs, such as the colon, a centerline can be manually marked by the radiologist while reviewing image data, such as CT, MRI or ultrasound data. However, because of the high level of manual intervention involved in this technique, it is expensive and generally not preferred.

An automatic method of generating a centerline is referred to as topological thinning, or "onion peeling." While there are numerous known variants of

BNSDOCID: <WO_____0229717A2_I_>

this technique, in general, volume units (voxels) from a region of interest are removed layer by layer from the boundary until what remains is a connected line of predetermined thickness, such as one voxel. While this technique has generally proven effective at determining a centerline in objects such as the colon, it has the disadvantage of being computationally expensive. In addition, with certain geometry's, the centerline that is determined by topological thinning is not always the centerline which would be intuitively desired.

Other techniques for generating a centerline through a virtual object include the use algorithms which attempt to determine the shortest path through an object, such as the Dijkstra algorithm, which is disclosed in the article "A Note on Two Problems in Connexion with Graphs," Dijkstra, Numerishe Mathemetik, vol. 1, pp 269-271, 1959, which is hereby incorporated by reference in its entirety. The Dijkstra algorithm finds the global minimal weight path in a weighted graph with nonnegative weights. This algorithm generally includes two phases: first, a distance from a source field (DSF) is created by labeling all graph vertices with the shortest distance from a single source to those vertices. Second, the shortest path is determined by tracing back from the farthest node to the source node. Thus, use of this algorithm requires mapping the image data to a graph form. An example of such a mapping is illustrated in Figure 8a. The Dijkstra algorithm has a draw back in the case of sharp turns occurring in the object. In such as case, the algorithm in seeking the shortest path tends to "hug the corner" and the centerline shifts from the center towards the boundary.

A skeleton of a virtual object can be defined in the context of a number of desired properties. As used herein, a skeleton should be tree-shaped and composed of single voxel paths through the branches. Thus, the skeleton represents a 1D curve in a 3D space rather than a 2D manifold. The skeleton should remain within the boundary of the region of interest and more preferably should remain in the "center" of the shape. For objects of arbitrary form, the concept of center may require a balancing between maximizing the distance from all of the boundaries of the boundaries of the region of interest and minimizing the extent that the centerline takes on a tortuous path. This balancing implies the desirability of applying a form of shortest path or union of shortest paths in deriving a skeleton.

3NSDOCID: <WO_____0229717A2_I_>

Although a number of techniques are known for generating a skeleton of an object, each of the known techniques have inherent drawbacks. Thus, there remains a need for improved techniques for generating centerlines, skeletons and or navigation paths within virtual objects.

The centerline is often used as the primary path for navigating through a virtual object. For example, in the case of virtual colonoscopy, the user is generally allowed to navigate along the centerline of the colon in either direction and can navigate off the centerline to explore regions of interest. Thus, virtual colonoscopy can provide the viewer with a more flexible view of the inner luminal surface of the colon than can be achieved with conventional optical colonoscopy. This is accomplished due to the fact that virtual reality and computer graphics allow the virtual camera to view from any arbitrary viewpoint and with any arbitrary view-direction, while optical colonoscopy is limited to a path in one direction along the colon (due to the fact that the optical camera is placed at the end of a probe with limited mobility). However, with the flexibility that is provided by virtual examination comes a new challenge. Specifically, the difficulty in following orientation changes. The colon is a very tortuous organ, and, as a result, when following along the colon in 3D users can become disoriented. This risk is especially present when the user veers off the centerline and examines a structure close to the colon wall. Often times, when returning to the centerline, the user may head back in the direction they previously came from. Since the view is from the opposite direction, it is usually not readily apparent that the flight is headed in the reverse direction. Thus it would be desirable to provide a method of reducing the opportunity for a viewer to become disoriented while navigating through the virtual object without reducing the desirable flexibility which can be achieved in the virtual environment.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide an improved method for generating a centerline, and where applicable a skeleton, of a virtual object.

In accordance with the present invention, a method for determining a centerline through a region of interest in a 3D image dataset is provided. The method includes identifying the boundaries of the region of interest and identifying the

BNSDOCID: <WO_____0229717A2_I_>

endpoints of the region of interest. For those points within the boundaries, a penalty value which is proportional to proximity of the point to a boundary is determined. A centerline is then identified by the longest Euclidean path connecting the endpoints which has the minimum penalized distance wherein the penalized distance reflects the actual accumulated pathlength and the penalties associated with the points along the path.

The method preferably includes steps to initially identify points which are near the centerline of the object. In this case, the steps of determining a penalty value and determining a path described above can be performed only on those points which are near the centerline. This reduces the dataset which is to be processed thereby improving processing cost and speed.

The operation of identifying points near the centerline can be accomplished by: computing a distance from boundary field (DBF) for each of the points in the region of interest; determining a gradient field for each point in the region of interest; identifying regions of points having a non-uniform gradient; and connecting the regions having a non-uniform gradient.

In addition to a single centerline, it is also desirable to extend the method to determine a complete skeleton of a branched object. This can be performed by identifying the endpoints of the branches extending from the centerline, and for each identified endpoint, determining a branch path connecting the endpoint to another branch which has the minimum penalized distance. Again, the penalized distance for each branch reflects the actual accumulated pathlength of the branch and the penalties associated with the points along the branch path.

BRIEF DESCRIPTION OF THE DRAWING

Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

Figure 1 is a flow chart illustrating a preferred method for generating a skeleton of a virtual object in accordance with the present invention;

BNSDOCID: <WO_____0229717A2_I_>

Figure 2 is a graphical representation of an outline of a region of interest along with a numerical representation of the Euclidean distance between each inside voxel and the object boundary or distance from boundary field (DBF);

Figure 3 is a graphical representation of gradient field vectors for voxels in a region of interest;

Figure 4 is a graphical representation illustrating the flagging of nonuniform gradient neighborhoods from the diagram of Figure 3.

Figure 5 is a flow chart illustrating a method for labeling nonuniform gradient vector field vectors;

Figure 6 is a flow chart for connecting flagged voxels of a centerline;

Figure 7 is a graphical representation illustrating the connection of flagged voxels along the gradient vector field vectors;

Figures 8a and 8b are a pictorial diagram illustrating a mapping of a voxel grid and neighbor relations to an undirected graph on which the Dijkstra algorithm can be applied, wherein Figure 8a represents a conventional shortest distance source field and Figure 8b represents a penalized distance from the source field, which is used in the present invention;

Figure 9a illustrates the generation of a centerline through an aorta;

Figure 9b illustrates the generation of a complete tree skeleton through an aorta, in accordance with the method of Figure 1.

Figure 10 is a cross section of a region of interest with a centerline and parameters for establishing the direction of travel along the centerline.

Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

BNSDOCID: <WO_____0229717A2_I_>

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention provides an improved and computationally efficient method of automatically generating a centerline in a lumen or a skeleton in a more complex object. The method uses the concept of penalized distance fields to augment known shortest path algorithms, such as the Dijkstra shortest path. The examples referred to are generally in the context of medical examination, such as virtual colonoscopy, but are applicable to any application of computer based virtual navigation.

As noted above, a problem with the known Dijkstra shortest path algorithm is evident when the path through the object turns sharply. In such a case, without some form of embellishment or modification, the Dijkstra shortest path will tend to hug the corner and deviate from the center of the object. In the present methods, this is corrected by use of the notion of a penalty field. Figure 8A illustrates a distance field connecting voxels in accordance with the conventional Dijkstra algorithm. Referring to Figure 8B, the present method supplements the neighbor edges of the Dijkstra algorithm by adding "penalty edges" which are weighted to provide an increased weight or penalty for those voxels near the object boundary. In the representation of Figure 8B, there are 27 vertices per voxel: one central vertex and 26 penalty vertices, that each share a penalty edge with the central vertex. The penalty edges have a weight equal to one half the penalty associated with including that voxel in the path. In this case, neighbor edges now connect to penalty vertices. This results in a graph that is a singly connected component where edges have positive weights to enable the Dijkstra algorithm to find the globally minimal shortest path. The cost associated with a path is the accumulated piecewise Euclidean distance of the path plus the sum of all of the penalties associated with the path. The centerline is defined as the minimum cost path through the penalized distance field.

Figure 1 is a flow chart depicting an method for determining a skeleton of a virtual object. The method assumes that image data of an object has been acquired, such as CT or MRI data, and that this data has transformed into the 3D volume domain and has been segmented. Acquiring 2D image data, such as CT, MRI, ultrasound, and the like and transforming the 2D data into a 3D representation having a plurality of voxels is well known in the art, as are various techniques for

BNSDOCID: <WO_____0229717A2_I_>

segmenting such 3D data. The data is thus presented as a 3D rectilinear grid which can be referred to as the volume wherein each volumetric sample point is called a voxel.

For the purposes of determining the centerline, the segmented image data can be subjected to a binary mask which delineates the object boundary and labels those voxels within the volume of the boundary as "inside" voxels. (step 102). The segmentation techniques disclosed in connection with electronic colon cleansing in U.S. patent application, serial number 09/777,120, filed on February 2, 2001 and entitled "System and Method for Performing a Three-Dimensional Virtual Examination of Objects, Such as Internal Organs," which is hereby incorporated by reference, can be used in this regard. With the boundary defined by the binary mask, the region of interest can be cropped using a bounding box which encloses those voxels tagged as "inside." (Step 104). While not essential to the practice of the invention, by cropping the data in step 104, a significant reduction in the data set is achieved and the subsequent steps are rendered more efficient. For example, in the case of medical imaging scans, the volume size is reduced on the order of 30-50%.

Following cropping, the Euclidean distance between each voxel flagged as "inside" the volume and the boundary is determined and recorded. (Step 106). As illustrated in Figure 2, this results in a mapping of the distance from boundary field (DBF). One method of determining the Euclidean distance is a four pass algorithm disclosed in the article "New Algorithms for Euclidean Distance Transformation of an n-Dimensional Digitized Picture with Applications," T. Saito et al., Pattern Recognition, vol. 27, no. 11, pp. 1551-1565. However, other algorithms for calculating the distance from the respective voxels to the boundaries can also be employed.

Preferably, for each voxel within the boundary of the region of interest, a central difference gradient is calculated to define a gradient vector field (GVF) of the distance from boundary fields (DBF). (Step 108). This can be achieved by comparing each voxel with only six neighboring voxels. The GVF at point (x,y,z) is found by $G_{xyz}=(G_x,G_y,G_z)$ where $G_x=V_{x+1,y,z}-V_{x-1,y,z}$ and $G_y=V_{x,y+1,z}-V_{x,y-1,z}$ and $G_z=V_{x,y,z+1}-V_{x,y,z-1}$. Higher numbers of neighboring voxels can be used in this calculation, such as by using a 26-voxel neighborhood Sobel filter, however

BNSDOCID: <WO_____0229717A2_I_>

increasing the number of voxels tends to slow the calculation without improving the resulting centerline. The graphical representation of Figure 3 illustrates a GVF for the representation of the DBF of Figure 2. In Figure 3, each gradient vector field vector associated with a respective voxel is illustrated as an arrow 300 with its origin at the voxel position.

After the GVF is established, the gradient vector field is analyzed to determine the characteristics of local neighborhoods within the gradient field vectors. The neighborhoods can be defined as 2x2x2 voxel cells. Referring to Figure 4, the neighborhoods can be characterized in six classes: local maxima near the centerline 402, local minima near the centerline 404, uniform areas near the centerline 410, uniform areas off the centerline 412, local maxima off the centerline 406 and local minima off the centerline 408.

Local maxima on or near the centerline 402 are defined as those regions near the centerline of the object in which all of the gradient vector field vectors of the 8 voxel neighborhood point in a direction towards the centerline. Thus, as illustrated by local maxima 414 and 416, all of the gradient vector field vectors are pointing in a different, but converging direction.

A local minima on the centerline 404 is defined as a neighborhood where all of the gradient vector field vectors surrounding such a minima are directed away from this point and towards a neighboring maxima. The vectors will then be in at least two groups of vectors pointing in opposite directions. In the case of a centerline through a lumen, each minima is bounded by a maxima on either side.

A local maxima located off of the centerline 406 generally has the same characteristics as a local maxima which is located on or near the centerline 402. The only difference being that such a local maxima is positioned remote from the centerline. This can occur in branches or distortions in the object off of the main centerline, such as illustrated by local maxima 418. Similarly, local minima off the centerline, such as local minima 420 are indistinguishable from local minima on the centerline, such as local minima 422, except for their position with respect to the centerline.

In addition to local maximas and local minimas there are also regions of uniform gradient vector field vectors, both on the centerline 410 and off the

BNSDOCID: <WO_____0229717A2_I_>

centerline 424. A uniform GVF is defined as one where the GVF vectors of the neighborhood, such as an eight voxel cell, have a direction which varies less than ninety degrees from the average direction. In those regions which are near the object boundary, the vectors in a local neighborhood generally point in the same direction which is perpendicular to the boundary. There can also be regions along the centerline where the object widens quickly. As a result, the GVF vectors neighboring the centerline in such regions will tend to point uniformly towards the center of this region. Thus, the method of labeling all nonuniform gradient vector field regions does not capture this portion of the centerline.

Given that a uniform GVF is defined as one where the GVF vectors of the neighborhood, such as an eight voxel cube or cell, have a direction which varies less than ninety degrees from the average direction, the process set forth in Figure 5 can be used to label local uniform and nonuniform GVF vectors. In step 505 the normalized average gradient field vector is computed for each defined neighborhood, such as 2x2x2 cells of voxels, for each voxel position. Next, the dot-product between each of the 2x2x2 GVF vectors and the associated average vector is determined and tested. (step 510). A positive dot product indicates a uniform GVF, whereas a zero or negative dot-product is indicative of a nonuniform GVF. Thus, to identify the nonuniform neighborhoods, those cells having a nonpositive dot product are labeled. (Step 515).

The use of a small number of voxels, i.e., 8, in each voxel position neighborhood allows for rapid execution time and results in the flagging of only a small percentage of voxels inside the boundary as being potentially near the centerline.

The result of the flagging process is a number of disconnected groups of flagged voxels. The next step is to connect the flagged voxels along the centerline (Figure 1, Step 110). An overview of a procedure for connecting the flagged voxels of the nonuniform GVF's is illustrated in Figures 6 and 7. First, a flagged voxel position, such as local minima 705, is selected as a starting point. (Step 605). Each vector of the flagged voxels has a component which points along the centerline and a component which points towards the centerline. Thus, a voxel by voxel path is traversed from the first flagged voxel, along the direction of the GVF vector (step

3NSDOCID: <WO_____0229717A2_I_>

610), until a next flagged voxel is reached (step 615). Referring to Figure 7, beginning with flagged voxel 710, the GVF vector is directed to voxel 712. This process is repeated until flagged voxel 714 of local maxima 716 is reached. Because the GVF vectors tend to point towards the centerline, the process of traversing along the voxels is a self correcting process. For example, voxel 710 is to the right of the centerline and the next voxel selected 712 is to the left of the centerline. Because the GVF vector of 712 is directed towards the centerline, the next voxel selected will at least be closer to the centerline or, perhaps, to the right of the centerline. The process of steps 605 through 615 is repeated for each group of flagged nonuniform neighborhoods.

Returning to Figure 1, a starting point voxel within the object is selected and the farthest voxel(s) from that voxel are determined by calculating the distance from the source field (DSF) (step 114). Such farthest voxels from any arbitrary voxel will represent the endpoints of the centerline and can be selected as a root voxel for the centerline or skeleton. The choice of starting point voxel can be improved by domain knowledge. In the case of colonoscopy, the rectum is the optimum point to start the examination, so we wish to select a point near the rectum. An estimation of the location of this point can be determined by finding the nearest "inside" point closest to the center-bottom of the dataset. This can be found by executing a search through the voxels near the center-bottom dataset voxel in shells of increasing radii until a voxel determined to be "inside" the colon is found.

In step 116, the penalized distance from the root voxel field (PDRF) is determined. Calculating the PDRF involves assigning a penalty to each voxel in the proposed path. The penalty, p at a voxel v is assigned based on the DBF value at that voxel calculated in step 106 as well as a global upper bound of all DBF values, M (M> max(DBF)). The penalty, p can be calculated as:

$$p(v) = 5000 \bullet [1 - \mathrm{DBF}(v)/M]^{16}$$

The term [1-DBF(v)/M] is always in the range of [0, 1] with the maximal values occurring for voxels near the boundary of the object. The term 5000, which is chosen heuristically, is selected to be sufficiently large to ensure that the resulting penalty is sufficiently large to overcome any distance advantages of a straight path. The value of 5000 allows skeleton segments to be up to 3000 voxels

BNSDOCID: <WO_____0229717A2_I_>

long without exceeding floating point precision. Other functions with similar characteristics can be chosen (e.g., small penalty towards the center and large at the boundary).

After assigning the penalty to each of the flagged voxels, the minimum cost path can be determined. (Step 118). The accumulated cost at each voxel along a path can be expressed as:

$$cost\ (v_k) = cost\ (v_k\text{-}1) + distance\ (v_k, v_{k\text{-}1}) + penalty\ (v_k)$$

In addition the accumulated cost, the accumulated distance at a given voxel, $D_a(v)$, can be expressed as:

$$D_a(v_k) = D_a(v_{k\text{-}1}) + \text{distance}\ (v_k, v_{k\text{-}1})$$

This accumulated distance can be used to identify the farthest voxel in the PDRF, which represents the opposite end of the centerline.

From the farthest voxel, the minimum cost path back to the source voxel is determined. This is performed, in a similar manner to the Dijkstra algorithm, by tracing the discrete centerline from the end voxel back to the root voxel. Because of the inclusion of higher penalties for voxels near the boundary, the minimum cost path tends to be pushed into the center of the object. This minimum cost path is the discrete centerline of the object. In the case of a tree like structure, this discrete centerline is the main branch of the discrete skeleton tree.

The discrete centerline is made up of numerous connected short straight line segments from voxel to voxel. Such a discrete "stair step" representation is not the most desirable centerline in many applications, such as virtual navigation. In such a case, the stair step can manifest itself as jitter in the virtual camera. Therefore, in many cases it is desirable to smooth the discrete centerline. (Step 120). One preferred method of smoothing the curve is to use an approximating spline algorithm with adaptive error tolerance. In narrow regions of the object, the error value should be small whereas in wider regions, more divergence from the discrete centerline can be tolerated. This error tolerance can be expressed as a percentage of the distance from the boundary. A percentage of less than 50% insures that the centerline remains closer to the centerline than to the boundary, thus it is guaranteed to be within the boundary. A well known B-spline curve which interpolates the first and last voxel and approximates the ones in between can be used with control points

BNSDOCID: <WO_____0229717A2_I_>

placed close to centerline voxels at nonuniform intervals. A number of heuristics can be used to minimize the number of control points required to acheive a particular accuracy. For example, an error tolerance of 35% may require 17% of the discrete centerline voxels as control points whereas an error tolerance of 50% would only require 13% of the discrete centerline voxels as control points.

In the case of a lumen- shaped object, such as a colon, where only a single centerline is desired, the smoothing operation would complete the centerline generation process. In the case of more complex, branched structures such as the lungs and aorta this initial centerline will serve as the main branch in the complete tree structure skeleton of the object.

The determination of the rest of the branches of the skeleton for a complex object are determined in steps 122 through 130 which are repeated for each branch of the skeleton.

The process of extending the centerline to a complete skeleton definition starts with the concept of "rolling an adaptive sphere" down the skeleton voxels. This, as explained below, will result in the labeling of those voxels which are near the centerline of the branches in the skeleton. (Step 130). The adaptive sphere is defined at each voxel along the skeleton in terms of its proximity to a boundary, a user applied scaling factor and a user defined constant. This is expressed mathematically as:

$$radius\ (v) = DistanceFromBoundary\ (v) \bullet scale + constant$$

Those voxels within the radius of the adaptive sphere are labeled as processed voxels. The combination of the terms *scale* and *const* determine the minimum feature size that will be used to determine a skeleton branch. For example, in a lumenal structure, such as the colon, a scale value of *scale*= 3 and *const*=50 results in the initial centerline labeling all colon voxels and thus remains the only centerline through the object. For branched objects, such as the aorta, the values *scale*=1.1 and *const* = 10 have been found to result in the identification of all significant vessels as branches of the skeleton.

To identify a branch along the centerline, a new penalized distance from root field (PDRF) is calculated for each of the voxels labeled as processed voxels in step 122. (Step 124). The PDRF is performed in a similar fashion as

3NSDOCID: <WO_____0229717A2_I_>

described above in connection with step 116. The voxel identified as the farthest voxel from the PDRF calculation is used as the tip of the next major branch.

From the tip voxel of the current branch, a path is determined along the PDRF for the current branch until an existing skeleton voxel is reached, i.e. the branches merge. (Step 126). As set forth above with respect to step 118, this involves identifying the minimum cost path through the PDRF of the branch voxels.

After the discrete minimum cost path is determined for the current branch, the branch centerline can be smoothed. (Step 128). The smoothing operation can be performed using the process described in connection with step 120 or any number of other curve smoothing techniques which are known. The one additional constraint that should be applied is that the smoothing operation should not result in a discontinuity at the point where two branches of the skeleton merge.

After the centerline for the branch is smoothed, those voxels of the branch centerline are added to the dataset of skeleton branches. (Step 130). Steps 122- 130 are repeated until the entire segmented volume of the object of interest has been analyzed.

Figure 9A illustrates the results performing steps 102 through 120 on a set of image data of an aorta. The aorta is rendered substantially transparent and the centerline 902 of the major branch of the aorta is highlighted. In Figure 9B, steps 120 through 130 have been iteratively repeated to extend centerline 902 into a full tree skeleton of the aorta in which all branch vessels have a centerline therethrough.

A further, optional method to improve the determination of both endpoints for any segment can be employed. The method is to compute the PDF from the approximate start point and find the maximum distance point from the approximate start point. The maximum distance point is then considered the true end point. Another PDF is computed from the true end point and the maximum distance point is then considered the true start point.

There may be more than one connected component (or segment) automatically identified as part of the colon. Some or all of these segments may actually be part of the colon. If more than two segments exists, the second segment may be selected by finding the nearest new segment voxel to the end of the first

3NSDOCID: <WO_____0229717A2_I_>

segment centerline. This point can be used as the starting point for the new PDRF in the new segment.

When two or more segments have been automatically identified by the system and the viewer disagrees with the assigned order, the user can be given the opportunity to select some or all of the segments (because some may be pockets of air in other organs not of interest). The selection can be made by clicking on each segment in the desired order of traversal. For example, in colonoscopy, the doctor usually clicks on the segment with the rectum first and the segment with the cecum last. Furthermore the user is able to reverse the preferred direction of travel within each segment so that the connection of all centerlines appears to be one continuous path. Furthermore, the selection of segments in order and the selection of directions within each segment can be accomplished by a single selection at one segment end or the other. The system can automatically determine which end the user selected and reverse the centerline direction if necessary.

In order to facillitate guided navigation throughout a skeleton, a user can define a start point and an endpoint for the navigation by selecting these points on the skeleton using a graphical user interface. If the start or end points are near the endpoints of the centerline or branch, selection need not be precise as the selected point can be snapped to the endpoint of the branch tips. If desired, new navigation fields can be calculated for this newly defined path and a new centerline can be generated between the selected points to reduce any transistion at branch junctions.

After a centerline of skeleton is determined through a virtual object, such a centerline is often used as a primary path for navigating through the object. As noted above, the user is generally provided with the opportunity to navigate in both directions along the centerline and can also navigate off of the centerline to explore regions of interest. However, with this flexibility in navigation comes the opportunity for the user to become disoriented.

To minimize the opportunity for disorientation while navigating in an interactive mode, the present method can employ a technique to color the centerline to indicate the direction of travel and to mark areas which have already been explored. The centerline is colored according to whether the user is traveling along the same direction as the current examination direction (i.e., rectum→cecum or

BNSDOCID: <WO_____0229717A2_I_>

cecum→rectum). As illustrated in Figure 10, the normal color (such as green) signifies that the viewpoint is within 135 degrees of the closest centerline direction. The opposite color (such as red) signifies that the viewpoint has come within 45 degrees of the opposite-direction of the closest centerline direction. The determination of this coloring is preferably filtered by a predetermined threshold value to add a level of hysterisis so that the centerline does not flash between the two colors (figure 1). The angle between the current view direction and the closest centerline direction is simply computed as the *arccos* of the *dot product* of the two directions. To be able to easily determine the closest centerline position during navigation, all valid positions for the virtual viewpoint store a pointer to the closest centerline point which is preferably defined by discrete points rather than as a continuous representation for quick lookup.

The PDF has another important use other than the creation of a centerline. The PDF can be used to translate the viewpoint of the user through each segment toward each segment centerline endpoint. The PDF field has the property that at every point the gradient simultaneously points away from the boundary and towards the segment centerline endpoint. By simply translating the viewpoint along the gradient direction until the viewpoint reaches the global minimum value, the viewpoint is smoothly drawn towards and along the centerline. This makes it easy for the user to efficiently traverse the colon without getting lost. Furthermore, the view direction can be rotated toward the direction of travel so that the user is looking where he is going. To smooth the transition from the current view direction to the new travel view direction, the view direction is combined with the new travel view direction with a weighted vector addition at each time step. Besides automatic navigation, the user can be permitted to freely navigate within the boundary of the object away from the centerline. The value of the DFB provides a measure of the virtual camera's proximity to the object boundary. If the viewpoint is within some threshold distance from the boundary, then a force can be activated which smoothly pushes the viewpoint "away from" the boundary. The "away from" direction can be estimated by using the gradient of the DFB. This gradient yields the most likely direction which points away from the boundary. The magnitude of the force can be

3NSDOCID: <WO_____0229717A2_I_>

determined by a function which is greatest closest to the boundary and decreases away from the boundary.

The cost field computation which employs penalties which are proportional to the proximity to the boundary of an object discussed above with respect to centerline generation can also be used in guided navigation through an object. Thus, the cost of a chosen navigation path can be the accumulated Euclidean distance of the chosen path plus the accumulated penalty cost of the path. Regardless of the starting point for a guided navigation, the penalty fields will tend to push the navigation towards the centerline and then direct the navigation smoothly down the centerline towards the selected seed point, which is typically chosen as an end point of the object.

The present methods can be performed on any number of computer processing platforms. It is expected that the increased efficiency that is provided by these techniques make them well suited for use on personal computer systems. For example, an IBM compatible Wintel PC operating under the Windows2000 operating system and having a single 1 GHZ CPU resulted in very reasonable processing times, such as generating a colon centerline less than two minutes, and the aorta skeleton of Figure 9B in 36 seconds.

Although the present invention has been described in connection with specific exemplary embodiments, it should be understood that various changes, substitutions and alterations can be made to the disclosed embodiments without departing from the spirit and scope of the invention as set forth in the appended claims.

BNSDOCID: <WO_____0229717A2_I_>

CLAIMS

1. A method for determining a centerline through a region of interest in a 3D image dataset comprising:

identifying the boundaries of the region of interest;

identifying the endpoints of the region of interest;

for those points within the boundaries, determining a penalty value which is a function of proximity of the point to a boundary;

determining a path connecting the endpoints which has the minimum penalized distance wherein the penalized distance reflects the actual accumulated pathlength and the penalties associated with the points along the path.

2. The method of claim 1, further comprising computing a distance from boundary field for each of the points in the region of interest.

3. The method of claim 2, further comprising identifying points which are near the centerline of the object and performing the steps of determining a penalty value and determining a path only on the points near the centerline.

4. The method of claim 3, wherein the operation of identifying point near the centerline further comprises:

determining a gradient field for each point in the region of interest;

identifying regions of points having a non-uniform gradient; and

connecting the regions having a non-uniform gradient.

5. The method of claim 1, further comprising smoothing the path connecting the endpoints.

6. The method of claim 1, wherein at least one of the endpoints is selected based on prior knowledge of the object.

7. The method of claim 1, further comprising:

identifying the endpoints of branches from the centerline;

3NSDOCID: <WO_____0229717A2_I_>

for each identified endpoint, determining a branch path connecting the endpoint to another branch which has the minimum penalized distance wherein the penalized distance reflects the actual accumulated pathlength and the penalties associated with the points along the branch path.

8. The method of claim 7, further comprising computing a distance from boundary field for each of the points in the branch paths.

9. The method of claim 7, further comprising, for each branch, identifying points which are near the centerline of the branch and performing the steps of determining a penalty value and determining a path only on the points near the centerline.

10. The method of claim 9, wherein the operation of identifying point near the centerline of the branch further comprises:

determining a gradient field for each point in the region of interest;

identifying regions of points having a non-uniform gradient; and

connecting the regions having a non-uniform gradient.

11. The method of claim 10, further comprising smoothing the centerline of each branch.

12. A method of generating a centerline through a virtual object comprising:

acquiring a 3D image dataset of the virtual object, the image dataset comprising a plurality of voxels;

identifying the boundaries of a region of interest in the 3D image dataset;

computing a distance from boundary field for each of the voxels in the region of interest;

determining a gradient voxel field for each voxel in the region of interest;

identifying regions of voxels having a non-uniform gradient;

connecting the regions of non-uniform gradient;

from the connected regions, identify a root voxel;

BNSDOCID: <WO_____0229717A2_I_>

for the connected voxels, calculate a penalized distance from the root voxel; and

determine the minimum penalized distance from the root voxel to the farthest voxel from the root voxel.

13. The method of claim 12, further comprising smoothing the path defined by the minimum penalized distance.

14. The method of claim 12, further comprising:

determining the endpoints of branches;

for each branch endpoint, determining a branch path connecting the endpoint to another branch which has the minimum penalized distance wherein the penalized distance reflects the actual accumulated pathlength and the penalties associated with the points along the branch path.

15. A method of guided navigation in a virtual object comprising:

identifying the boundaries of the region of interest;

for those points within the boundaries, determining a penalty value which is proportional to proximity of the point to a boundary;

controlling at least one of the speed and direction of a virtual camera in accordance with the penalty value of the current camera position.

16. The method of guided navigation of claim 15, wherein the penalty field is a distance from boundary field.

17. A method of guided navigation along a centerline of a virtual object comprising:

determining an initial navigation direction along the centerline;

marking the centerline with a first indicia as navigation proceeds along the initial direction of navigation; and

marking the centerline with a second indicia as navigation proceeds in a direction substantially opposite to the first direction.

3NSDOCID: <WO_____0229717A2_I_>

18. The method of claim 17, wherein the first indicia is a first color and the second indicia is a second color different from the first color.

19. The method of claim 17 wherein the initial navigation direction is defined as a direction within about plus or minus 70 degrees of the centerline.

20. The method of claim 19, wherein the object is a colon and the initial direction of navigation is from rectum to secum.

21. The method of claim 19, wherein the object is a colon and the initial direction of navigation is from secum to rectum.

BNSDOCID: <WO_____0229717A2_I_>

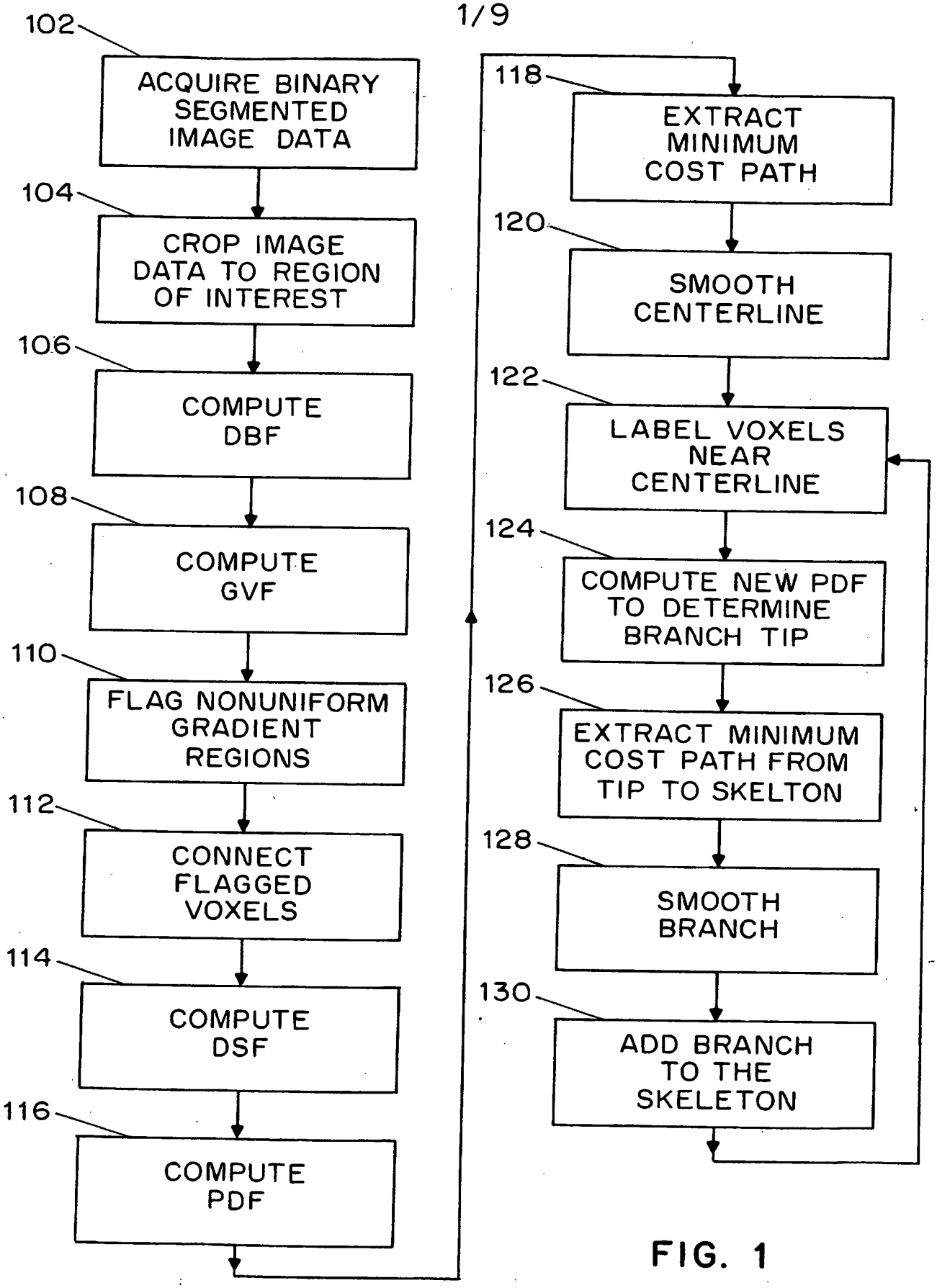


FIG. 1

BNSDOCID: <WO_____0229717A2_I_>

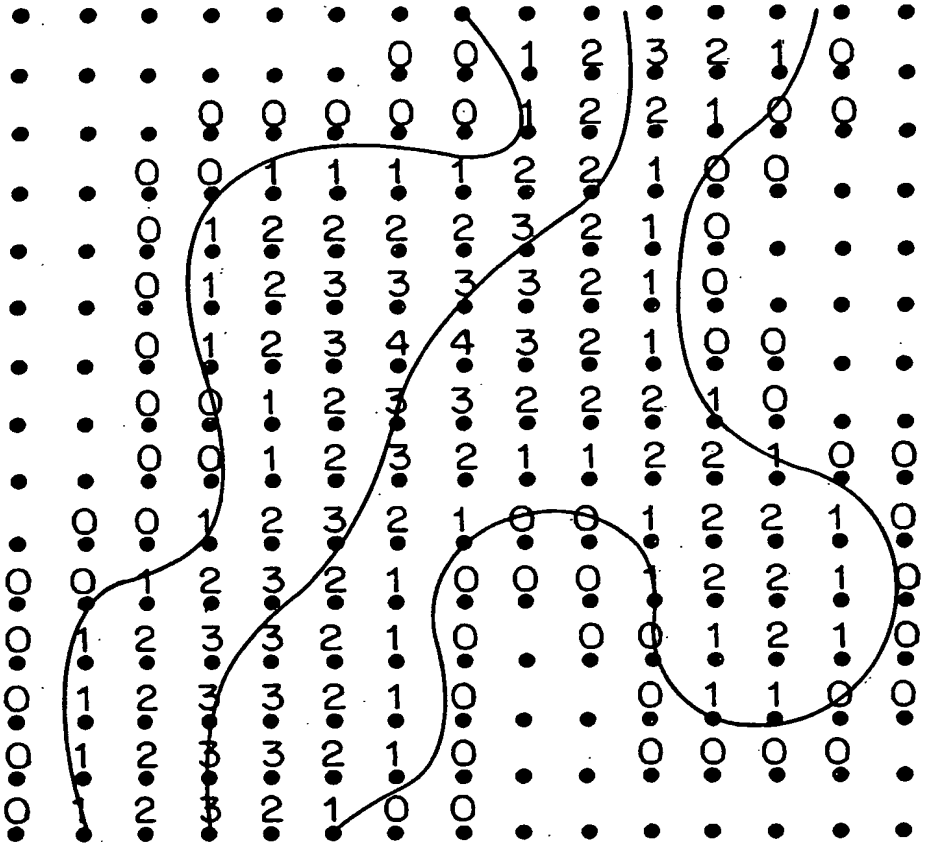


FIG. 2

BNSDOCID: <WO_____0229717A2_I_>

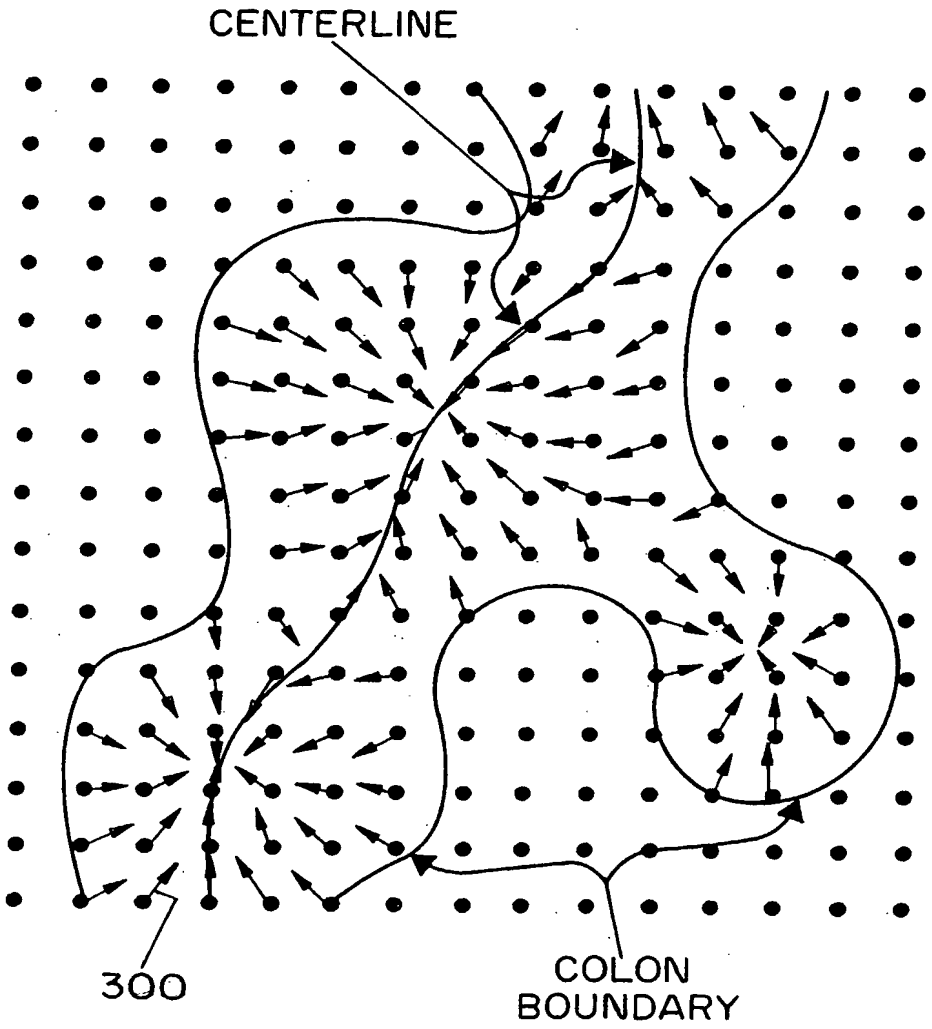
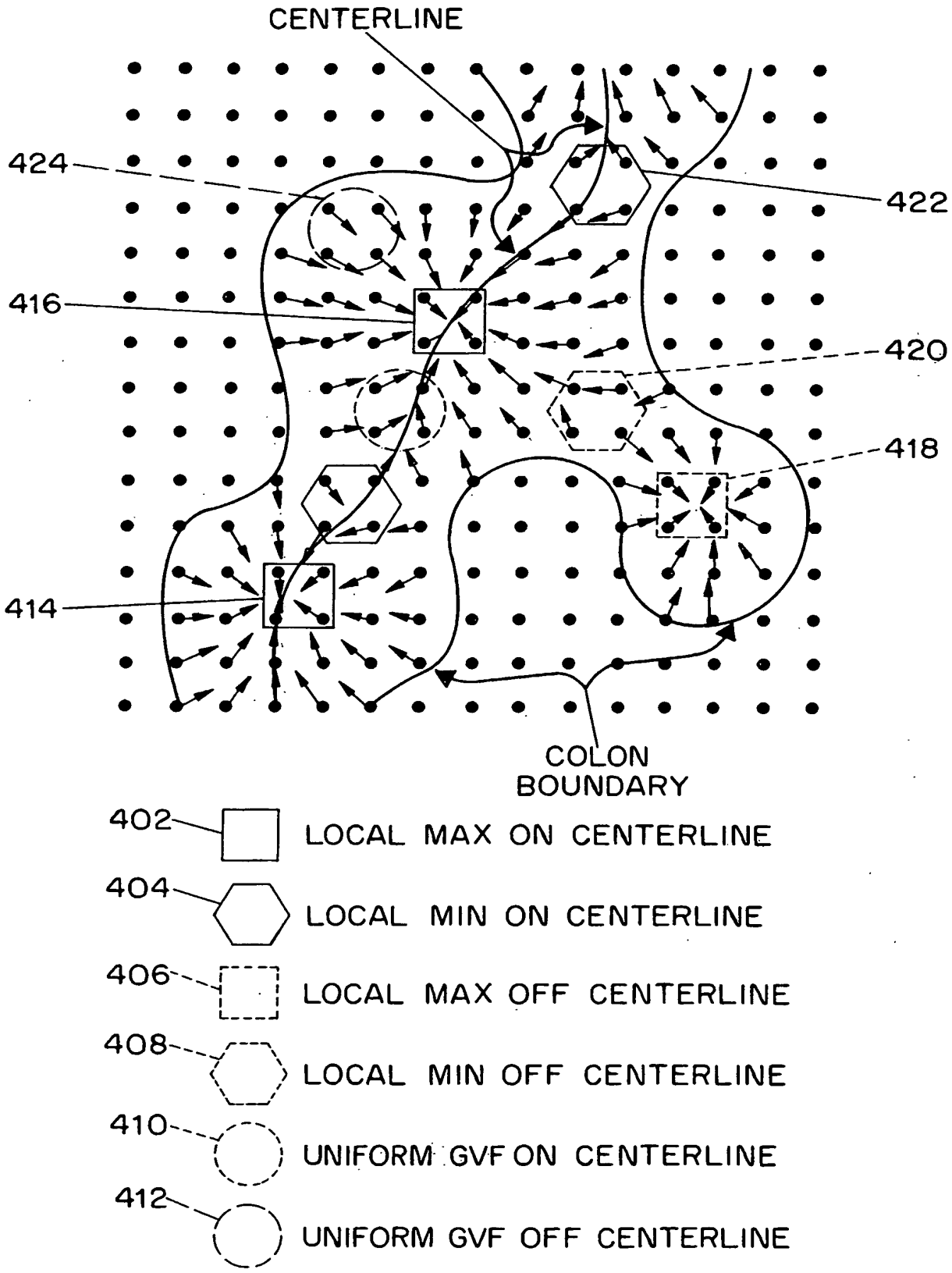


FIG. 3

FIG. 4


CENTERLINE
424
422
416
420
418
414
COLON BOUNDARY
402
LOCAL MAX ON CENTERLINE
404
LOCAL MIN ON CENTERLINE
406
LOCAL MAX OFF CENTERLINE
408
LOCAL MIN OFF CENTERLINE
410
UNIFORM GVF ON CENTERLINE
412
UNIFORM GVF OFF CENTERLINE

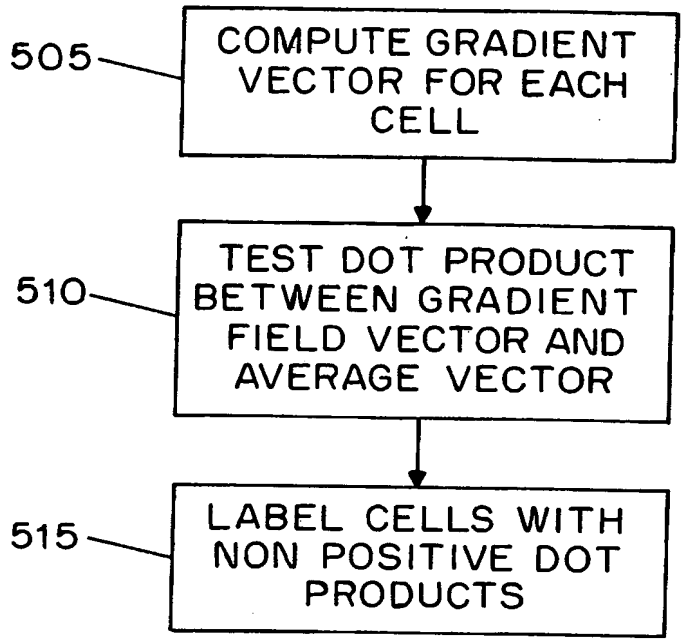

505
COMPUTE GRADIENT VECTOR FOR EACH CELL
510
TEST DOT PRODUCT BETWEEN GRADIENT FIELD VECTOR AND AVERAGE VECTOR
515
LABEL CELLS WITH NON POSITIVE DOT PRODUCTS

FIG. 5

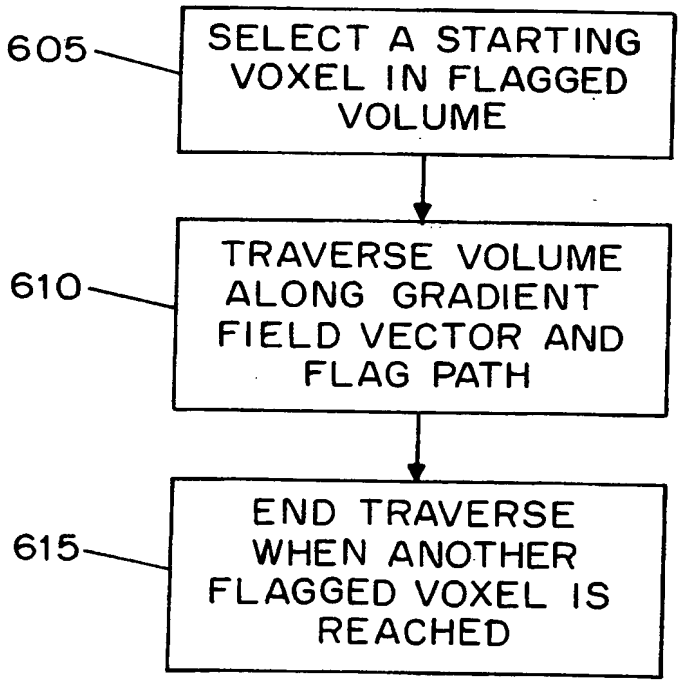

605
SELECT A STARTING VOXEL IN FLAGGED VOLUME
610
TRAVERSE VOLUME ALONG GRADIENT FIELD VECTOR AND FLAG PATH
615
END TRAVERSE WHEN ANOTHER FLAGGED VOXEL IS REACHED

FIG. 6

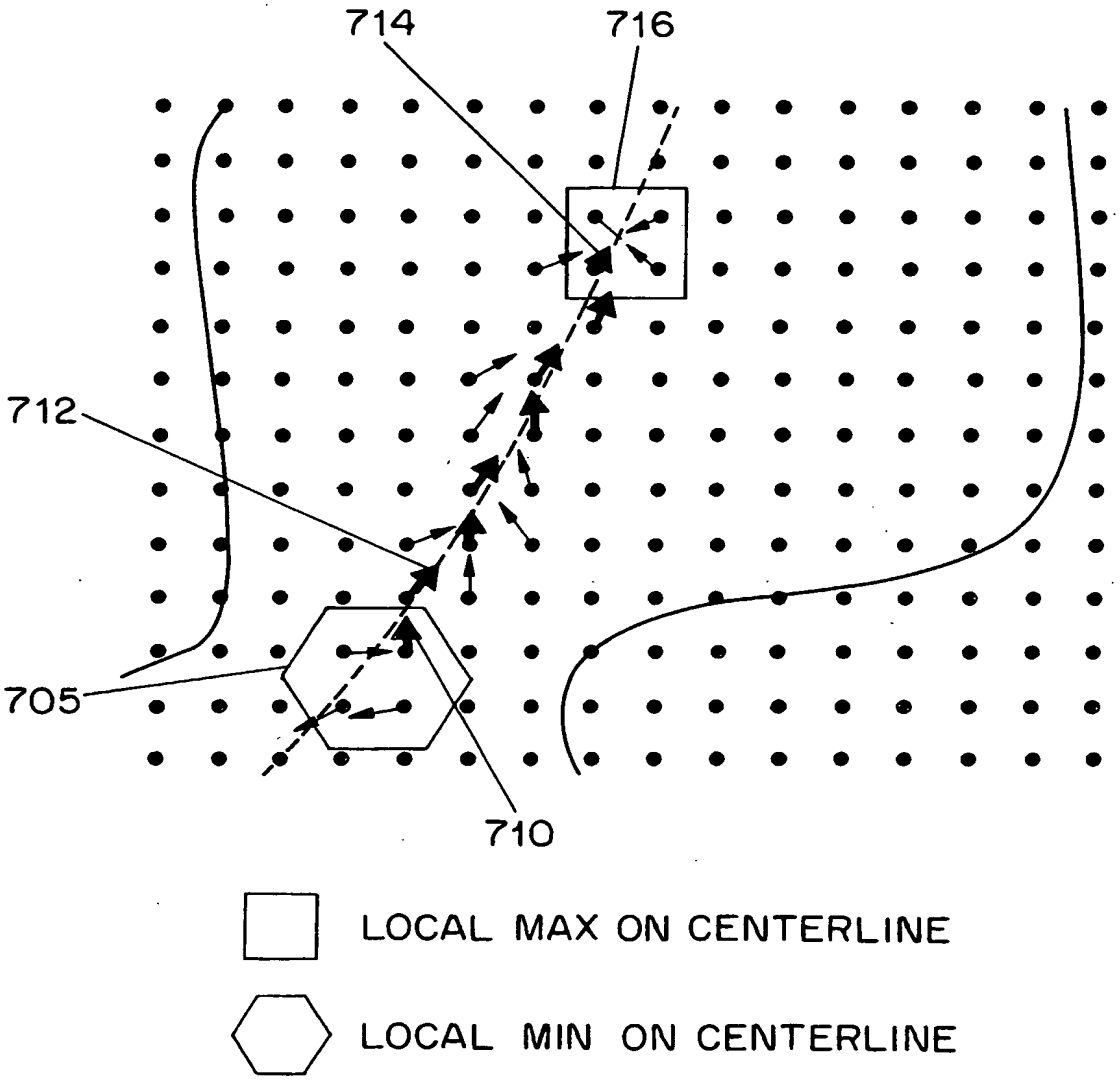
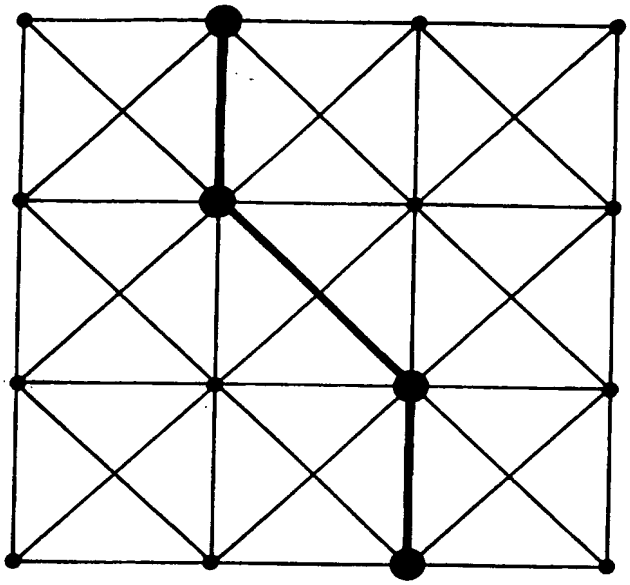

714
716
712
705
710
LOCAL MAX ON CENTERLINE
LOCAL MIN ON CENTERLINE

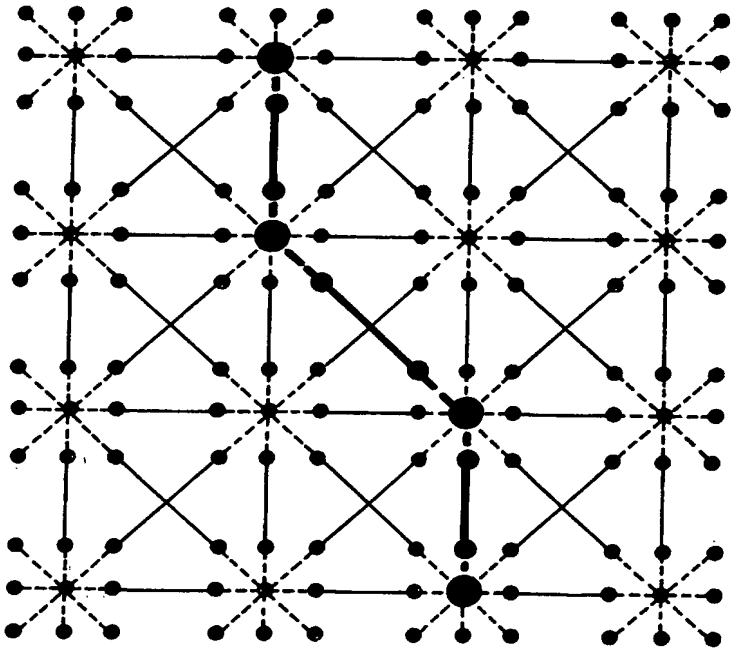
FIG. 7



NEIGHBOR EDGE: ———————

LOWEST COST PATH: ●———●

FIG. 8A



PENALTY EDGE: - - - - - - - - - - - - - - - - -

LOWEST COST PATH: ●-●-●———●-●-●

FIG. 8B

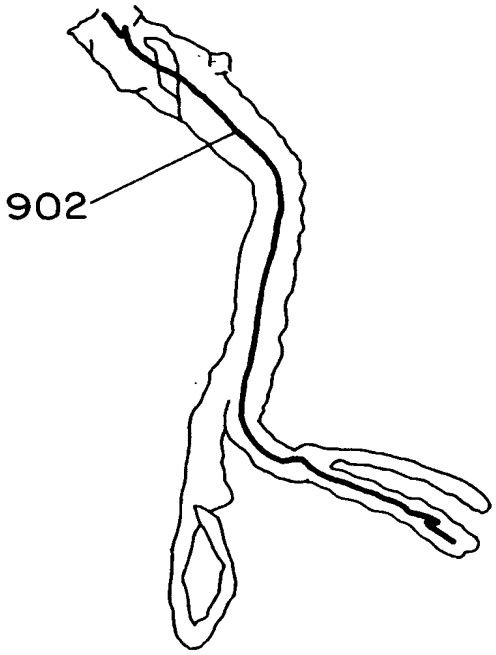


FIG. 9A

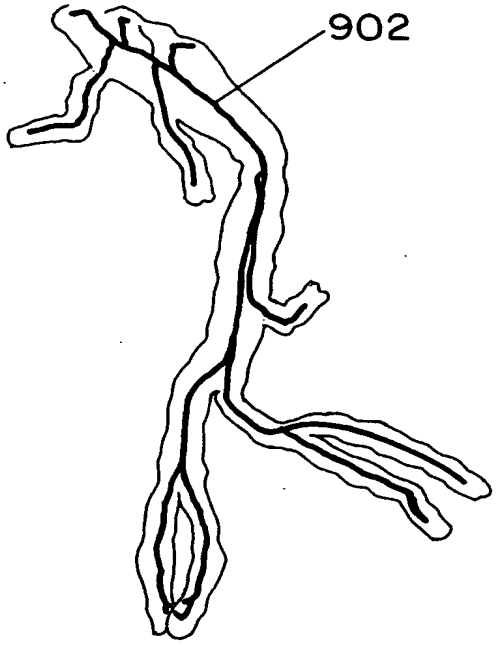


FIG. 9B

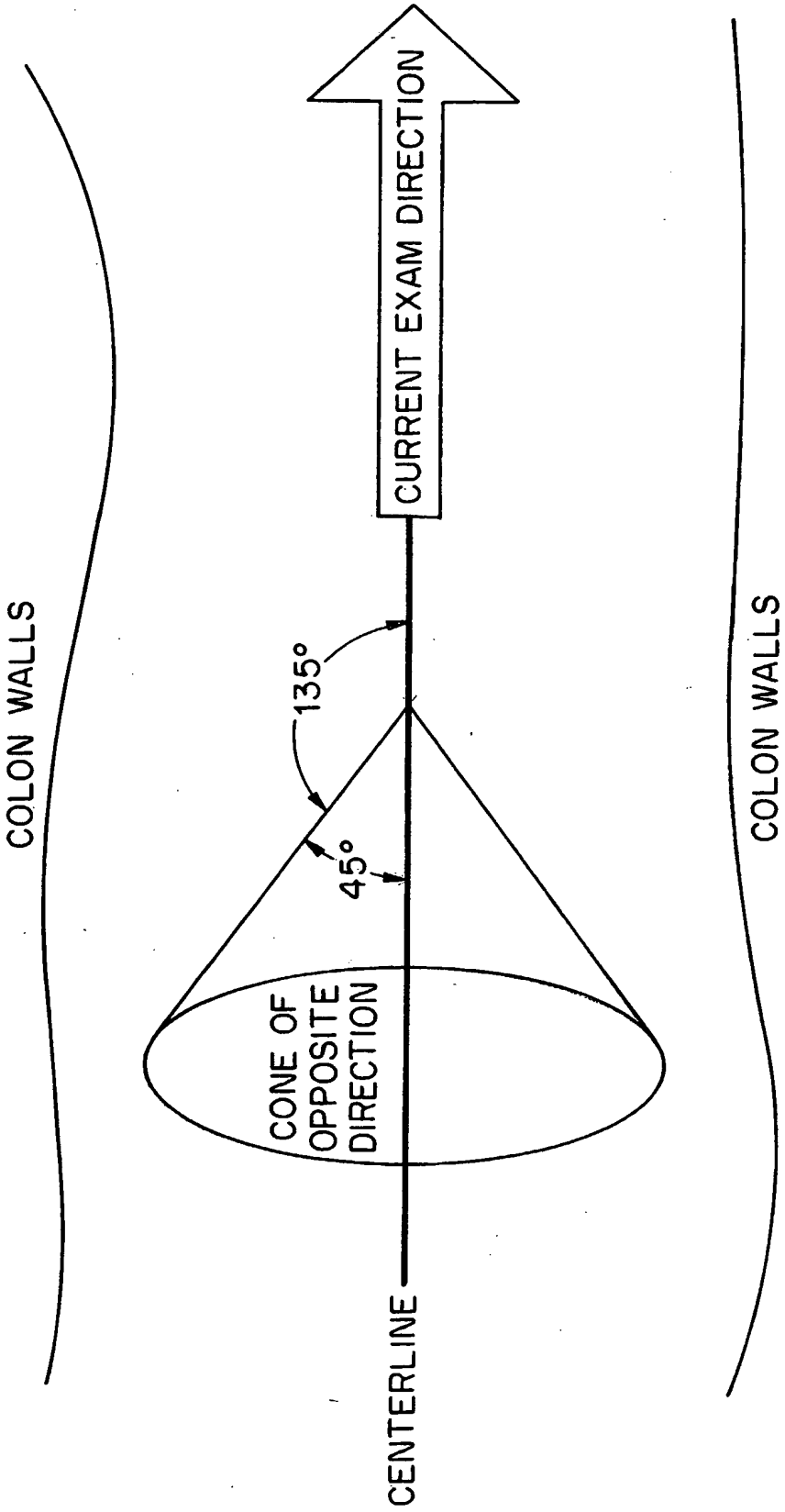


FIG. 10

BNSDOCID: <WO_____0229717A2_I_>